

Laboratorio Nro. 5

Implementación de Grafos



Objetivos: 1. Comparar las ventajas y desventajas de implementaciones dinámicas y estáticas de estructuras de datos. 2. Escoger la estructura de datos apropiada para resolver un problema dado. 3. Resolver problemas fundamentales de grafos, incluyendo búsqueda DFS y BFS



Consideraciones: Lean y verifiquen las consideraciones de entrega,



Leer la Guía



Trabajo en Parejas



Si tienen reclamos,
regístenlos en
<http://bit.ly/2g4TTKf>



Ver calificaciones en Eafit Interactiva



En el GitHub docente, encontrarán la traducción de los Ejercicios en Línea

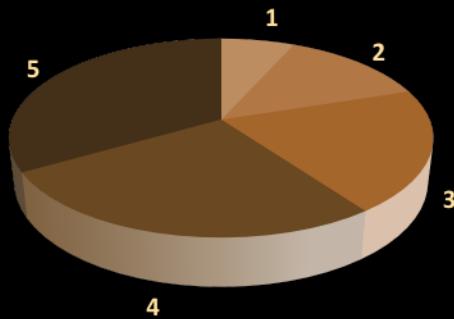


Hoy, plazo de entrega



Subir el informe pdf en la carpeta **informe**, el código del ejercicio 1 en la carpeta **codigo** y el código del 2 en la carpeta **ejercicioEnLinea**

Porcentajes y criterios de evaluación



- 1. Simulacro sustentación proyecto
- 2. Análisis de complejidad
- 3. Código de laboratorio
- 4. Simulacro de parcial
- 5. Ejercicio con juez en línea

1. Simulacro de Proyecto

Códigos para entregar en GitHub en la carpeta código

1 Simulacro de Proyecto

Códigos para entregar en GitHub en la carpeta **codigo**:



Vean Guía
numeral 3.4



Código del ejercicio en línea
en **GitHub**. Vean Guía en
numeral 4.24



Documentación
opcional. Si lo
hacén, utilicen
Javadoc o
equivalente. No
suban el HTML a
GitHub.



No se reciben archivos
en **.RAR** ni en **.ZIP**



Utilicen Java, C++ o Python

Los vehículos eléctricos son una de las tecnologías más promisorias para reducir la dependencia del petróleo y las emisiones de gases invernadero. El uso de vehículos eléctricos para carga y para el transporte de pasajeros tiene una limitación:

El rango de conducción es limitado y el tiempo de carga de la batería es relativamente alto. Por esta razón, es necesario considerar que los vehículos se desvíen de la ruta para ir a estaciones donde puedan recargar su batería



Un ejemplo de carro eléctrico es Kratos, desarrollado por EAFIT. Foto de <https://bit.ly/2SaNotS>

Texto tomado de <https://goo.gl/AvWs6B>

PhD. Mauricio Toro Bermúdez

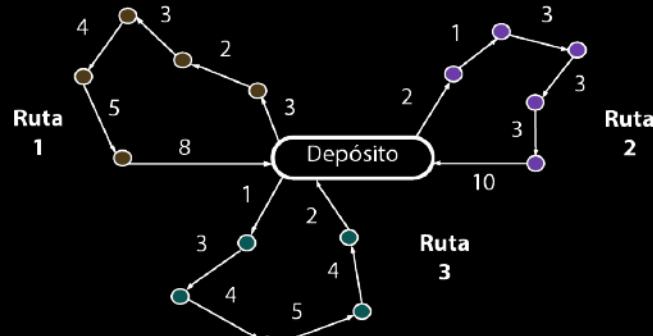
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS1

Código ST0245

Un problema que requiere una urgente solución es cómo encontrar las rutas óptimas para que un conjunto de vehículos eléctricos reparta mercancía a un conjunto de clientes. Dado una lista de clientes ubicados en un mapa vial bidimensional, un depósito de inicio y fin, y restricciones como la autonomía de la batería, la duración máxima de una ruta.

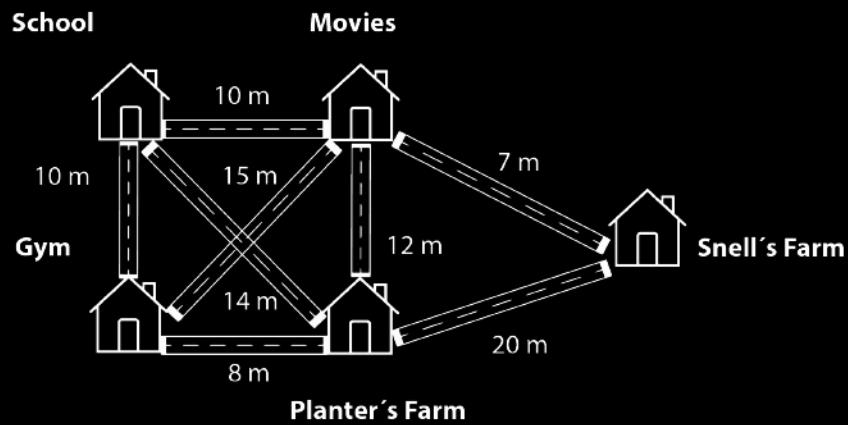
La solución a este problema debe responder la pregunta ¿cuáles son las rutas para una flota de vehículos eléctricos, para visitar todos los clientes de una empresa, minimizando el tiempo total? El tiempo total es la suma del tiempo del recorrido, el tiempo de visitar a los clientes y el tiempo que toman las recargas de batería. El primer paso solucionar este problema es definir qué estructura de datos se utilizará para representar el mapa de una ciudad



► **Implementen un algoritmo que permita crear una estructura de datos con el mapa de una ciudad. Para realizar una prueba, en *Github* encontrarán el archivo *medellin_colombia-grande.txt* que contiene un mapa de la ciudad de Medellín**



Ejemplo 1, para el siguiente mapa, el archivo de entrada es el siguiente:



ESTRUCTURA DE DATOS1
Código ST0245

Vertices. Formato: ID, coordenada x, coordenada y, nombre

```
10000 2.00000 0.00000 School
1 4.00000 1.00000 Movies
2 5.00000 2.00000 Snell
3 2.00000 5.00000 Planters
4 0.00000 2.00000 Gym
```

Arcos. Formato: ID, ID, distancia, nombre

```
10000 1 10.0 Calle 1
10000 3 14.0 desconocido
10000 4 10.0 desconocido
1 10000 10.0 Calle 2a
1 2 7.0 desconocido
1 3 12.0 desconocido
1 4 15.0 desconocido
2 1 7.0 desconocido
2 3 20.0 desconocido
3 10000 14.0 desconocido
3 1 12.0 desconocido
3 2 20.0 desconocido
3 4 8.0 desconocido
4 10000 10.0 desconocido
4 1 15.0 desconocido
4 3 8.0 desconocido
```

2. Simulacro de Maratón de Programación sin documentación HTML, en GitHub, en la carpeta ejercicioEnLinea

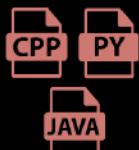
2 Simulacro de Maratón de Programación sin documentación HTML, en GitHub, en la carpeta ejercicioEnLinea



Vean Guía numeral 3.3



No se requiere documentación para los ejercicios en línea



Utilicen Java, C++ o Python



No se reciben archivos en .PDF ni .TXT



No se reciben archivos en .RAR ni en .ZIP



Código del ejercicio en línea en **GitHub**. Vean Guía en numeral 4.24

2.1 Resuelvan el siguiente ejercicio

En 1976, el teorema de colorear un mapa con 4 colores fue probado con la ayuda de un computador. Este teorema muestra que un mapa puede ser coloreado solamente con 4 colores, de tal forma que no haya una región coloreada usando el mismo color que un vecino. Aquí hay un problema similar a ese problema, pero es mucho más simple.



Ustedes tienen que decidir si dado un grafo conexo arbitrario, ese grafo se puede colorear con 2 colores. Esto quiere decir, si uno puede asignar colores (de una paleta de 2 colores) a los nodos, de tal forma que no haya 2 nodos adyacentes del mismo color. Para simplificar el problema ustedes pueden asumir que:

- No hay un nodo que tenga un arco a sí mismo.
- El grafo es no dirigido, es decir que, si un nodo a está conectado a un nodo b, usted puede asumir que el nodo b también está conectado al nodo a.

ESTRUCTURA DE DATOS1

Código ST0245

- c. El grafo será fuertemente conexo. Esto quiere decir, que hay al menos un camino de un nodo de grafo a cualquier otro nodo.



La entrada consiste en varios casos de prueba. Cada caso de prueba comienza con una línea que tiene un número n ($1 < n < 200$) de nodos diferentes. La siguiente línea contiene el número de arcos.

Posteriormente, las siguientes líneas, cada una contiene 2 números que especifican que existe un arco entre dos nodos.

Un nodo en el grafo se representa con un número a ($0 < a < n$). Una entrada con $n = 0$ simboliza el fin de la entrada y no debe ser procesada.



Ustedes tienen que decidir si el grafo de entrada puede ser coloreado con dos colores o no, y deben imprimirlo como se muestra a continuación.



```

3
3
0 1
1 2
2 0
3
2
0 1
1 2
9

```

ESTRUCTURA DE DATOS1
Código ST0245

8
0 1
0 2
0 3
0 4
0 5
0 6
0 7
0 8
0



Ejemplos de la salida

NOT BICOLORABLE.
BICOLORABLE.
BICOLORABLE.



[Opc] Para los numerales 2.2 al 2.5, resuelvan los siguientes problemas

2.2 <http://bit.ly/2gTLZ53>

2.3

<http://bit.ly/2hGqJPB>

2.4

<http://bit.ly/2hrrCfS>

2.5

<http://bit.ly/2k8CGSG>

3. Simulacro de preguntas de sustentación de Proyecto en la carpeta informe

3 Simulacro de preguntas de sustentación de Proyecto en la carpeta informe



Vean **Guía** numeral 3.4



Exporten y entreguen informe de laboratorio en **PDF, en español o Inglés**



Si hacen el **informe en español**, usen la **plantilla en español**



No apliquen **Normas ICONTEC** para esto

Si hacen el **informe en inglés**, usen a **plantilla en inglés**

Sobre el Simulacro de Proyectos

3.1



Escriban una explicación entre 3 y 6 líneas de texto **del código del numeral 1**. Digan cómo funciona, cómo está representado el mapa de la ciudad, por ejemplo, utilizaron matrices, listas, tablas de hash, ¿por qué? Pueden apoyarse realizando una imagen para facilitar su explicación.

3.2



Nota: La explicación del código debe ir dentro del informe y no dentro del código

3.2



Si representamos el mapa de Medellín del numeral 1 con matrices de adyacencia, ¿Cuánta memoria consumiría? Tengan en cuenta que hay alrededor de 300,000 vértices.



3.3 ¿Cómo solucionaron el problema de que los identificadores de los puntos del mapa no empiezan en cero?

Sobre el simulacro de maratón de programación



3.4 Expliquen con sus propias palabras la estructura de datos que utilizan para resolver el problema, y cómo funcionan los algoritmos realizados en el numeral 2.1 y los ejercicios opcionales que hayan hecho del punto 2. Esto en 3 a 6 líneas de texto.



3.5 Calculen la complejidad del ejercicio 2.1 y, si los hicieron, de los Ejercicios Opcionales.



3.6 Expliquen con sus palabras las variables (*qué es 'n'*, *qué es 'm'*, etc.) del cálculo de complejidad del numeral 3.5. Ver ejemplo a continuación:



Ejemplo de esta respuesta:

“n es el número de elementos del arreglo”,
“V es el número de vértices del grafo”,
“n es el número de filas de la matriz y m el número de columnas’.

4. Simulacro de parcial en informe PDF

4 Simulacro de Parcial en el informe PDF

Resuelvan los ejercicios



Para este simulacro,
agreguen **sus respuestas**
en el informe PDF.



*El día del Parcial no
tendrán computador,
JAVA o acceso a internet.*



Si hacen el **informe en
español**, usen la
plantilla en español



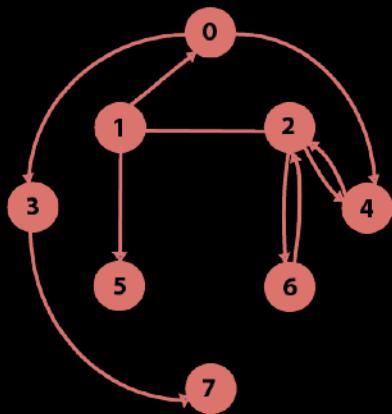
Exportar y entregar informe
de laboratorio en **PDF**, en
español o inglés

Si hacen **el informe en
inglés**, usen a **plantilla
en inglés**



No apliquen **Normas Icontec** para esto

4.1 [Opc] Consideren el no hay arco, por simplicidad, deje el espacio en blanco.



	0	1	2	3	4	5	6	7
0					1	1		
1								
2								
3								
4								
5								
6								
7								

4.2

Para el mismo grafo, completen la representación de **listas de adyacencia**. Como el grafo no tiene pesos, sólo se colocan los sucesores en la lista de adyacencia.

0 ->[3,4]

1 ->

2 ->

3 ->

4 ->

5 ->

6 ->

7 ->

4.3

¿Cuánta memoria (ojo, no tiempo sino memoria) ocupa una representación usando listas de adyacencia para un grafo dirigido con n vértices en el peor de los casos?

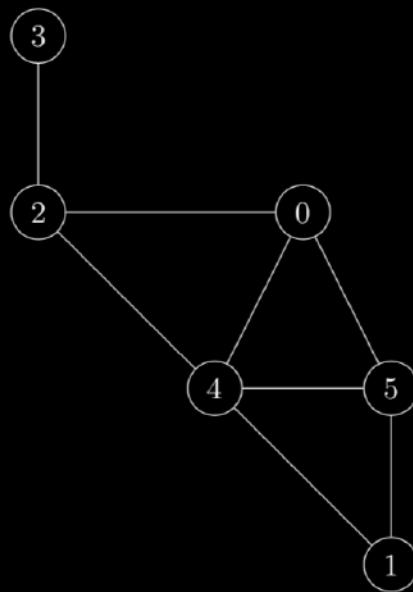
- a)** $O(n)$
- b)** $O(n^2)$
- c)** $O(1)$
- d)** $O(\log n)$
- e)** $O(n \cdot \log n)$

4.4

Se tiene el siguiente grafo no dirigido:

ESTRUCTURA DE DATOS1

Código ST0245



4.4.1 ¿Cuál es un recorrido de *búsqueda primero en profundidad* del grafo anterior, si como nodo inicial se toma el nodo 1?

- i) 1, 5, 0, 3, 2, 4
- ii) 1, 4, 5, 0, 2, 3
- iii) 1, 4, 0, 3, 5, 2
- iv) 1, 5, 4, 0, 3, 2

4.4.2 ¿Cuál es un recorrido de *búsqueda primero en amplitud* del grafo anterior, si se toma como nodo inicial el nodo 1?

- i) 1, 4, 5, 0, 2, 3
- ii) 1, 5, 0, 2, 3, 4
- iii) 1, 4, 2, 0, 3, 5
- iv) 1, 3, 0, 4, 5, 2

ESTRUCTURA DE DATOS1**Código ST0245****PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas

Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627

Tel: (+57) (4) 261 95 00 Ext. 9473



5. [Opcional]

Lecturas Recomendadas

5 [Opc] Lecturas recomendadas



Vean Guía en
numeral 3.5 y 4.20



Exportar y entregar informe
de laboratorio en **PDF, en
español o Inglés**



Si hacen el **informe
en español**, usen la
plantilla en español



No apliquen **Normas
Icontec** para esto

Si hacen el **informe
en inglés**, usen a
plantilla en inglés



"El ejercicio de una profesión requiere la adquisición de competencias que se sustentan en procesos comunicativos. Así cuando se entrevista a un ingeniero recién egresado para un empleo, una buena parte de sus posibilidades radica en su capacidad de comunicación; pero se ha observado que esta es una de sus principales debilidades..." Tomado de <https://aprendeonline.udea.edu.co/revistas/index.php/ingeso/article/viewFile/13986/12398>



Lean a “*Robert Lafore, Data Structures and Algorithms in Java (2nd edition), Chapter 13: Graphs. 2002*” y sumen puntos adicionales, así:



Hagan un mapa conceptual con los principales elementos teóricos.

Otras sugerencias de lectura

Si desean otras lecturas, consideren las siguientes:



Thomas Cormen, *Introduction to Algorithms* (3th edition), Sections 23.2, 23.3 y 23.5. 2009, que pueden encontrar en biblioteca



“John Hopcroft et al., *Estructuras de Datos y Algoritmos, Capítulo 7: Grafos no dirigidos. 1983*” que pueden encontrar en biblioteca.

6. [Opcional]

Trabajo en Equipo y Progreso Gradual

6 [Opc] Trabajo en Equipo y Progreso Gradual



Vean Guía en **numeral 3.5 y 4.20**



Si hacen el **informe en español**, usen la **plantilla en español**



Si hacen el **informe en inglés**, usen a **plantilla en inglés**



Exportar y entregar informe de laboratorio en **PDF, en español o Inglés**



No apliquen **Icontec** para esto

Normas



Entreguen copia e integrantes que participaron



Entreguen el reporte de *git* con los cambios en el código y quién hizo cada cambio, con fecha, hora e integrantes que participaron



Entreguen el reporte de cambios del informe de laboratorio que se genera *Google docs* o herramientas similares



Nota: Estas respuestas también deben incluirlas en el informe PDF

7. [Opcional]

Laboratorio en Inglés con

plantilla en Inglés

7 [Opc] Laboratorio en inglés



Vean Guía en
numeral 3.5 y 4.20



Si hacen el **informe en español**, usen la
plantilla en español



El inglés es un idioma importante en la Ingeniería de Sistemas porque
la mayoría de los avances en tecnología se publican en este idioma y la
traducción, usualmente se demora un tiempo.

Adicionalmente, dominar el inglés permite conseguir trabajos en el
exterior que son muy bien remunerados. *Tomado de goo.gl/4s3LmZ*



Exportar y entregar informe
de laboratorio en **PDF, en
español o Inglés**



No apliquen **Normas
Icontec** para esto



Entreguen el código y el informe en inglés.

Resumen de ejercicios a resolver

1.1 Implementen un algoritmo que permita crear una estructura de datos con el mapa de una ciudad. Para realizar una prueba, en *Github* encontrarán el archivo *medellin_colombia-grande.txt* que contiene un mapa de la ciudad de Medellín

2.1 Decidir si dado un grafo conexo arbitrario, ese grafo se puede colorear con 2 colores. Esto quiere decir, si uno puede asignar colores (de una paleta de 2 colores) a los nodos, de tal forma que no haya 2 nodos adyacentes del mismo color.

2.2 [Ejercicio Opcional] <http://bit.ly/2gTLZ53>

2.3 [Ejercicio Opcional] <http://bit.ly/2hGqJPB>

2.4 [Ejercicio Opcional] <http://bit.ly/2hrrCfS>

2.5 [Ejercicio Opcional] <http://bit.ly/2k8CGSG>

3.1 Escriban una explicación entre 3 y 6 líneas de texto **del código del numeral 1**. Digan cómo funciona, cómo está representado el mapa de la ciudad, por ejemplo, utilizaron matrices, listas, tablas de hash, ¿por qué? Pueden apoyarse realizando una imagen para facilitar su explicación

3.2 Si representamos el mapa de Medellín del numeral 1 con matrices de adyacencia, ¿Cuánta memoria consumiría? Tengan en cuenta que hay alrededor de 300,000 vértices

3.3 ¿Cómo solucionaron el problema de que los identificadores de los puntos del mapa no empiezan en cero?

3.4 Expliquen con sus propias palabras la estructura de datos que utilizan para resolver el problema, y cómo funcionan los algoritmos realizados en el numeral 2.1 y los ejerciciosopcionales que hayan hecho del punto 2. Esto en 3 a 6 líneas de texto.

3.5 Calculen la complejidad del ejercicio 2.1 y, si los hicieron, de los Ejercicios Opcionales

3.6 Expliquen con sus palabras las variables (*qué es 'n'*, *qué es 'm'*, etc.) del cálculo de complejidad del numeral 3.5

4. Simulacro de Parcial

5. [Ejercicio Opcional] Lectura recomendada

ESTRUCTURA DE DATOS1
Código ST0245

6. [Ejercicio Opcional] Trabajo en Equipo y Progreso Gradual

7. [Ejercicio Opcional] Utilicen la plantilla dispuesta en este idioma para el laboratorio.

Ayudas para resolver los Ejercicios

Ayudas para el Ejercicio 1.....	<u>Pág. 26</u>
Ayudas para el Ejercicio 2.1.....	<u>Pág. 26</u>
Ayudas para el Ejercicio 2.2.....	<u>Pág. 27</u>
Ayudas para el Ejercicio 2.3.....	<u>Pág. 27</u>
Ayudas para el Ejercicio 2.4.....	<u>Pág. 27</u>
Ayudas para el Ejercicio 3.4.....	<u>Pág. 27</u>
Ayudas para el Ejercicio 3.5.....	<u>Pág. 28</u>
Ayudas para el Ejercicio 5.....	<u>Pág. 28</u>
Ayudas para el Ejercicio 6.1.....	<u>Pág. 28</u>
Ayudas para el Ejercicio 6.2.....	<u>Pág. 29</u>
Ayudas para el Ejercicio 6.3.....	<u>Pág. 29</u>



Ayudas para el Ejercicio 1



Pista 1: Vean en **numeral 4.13** “Cómo usar Scanner o BufferedReader”



Pista 2: Hay información que sobra, por ejemplo, la latitud y la longitud de cada vértice y el nombre de cada arista.



Pista 3: Es mejor usar *BufferedReader* porque es más rápido que *Scanner*. La idea es leer en una cadena de caracteres el contenido de cada línea y usando el método *split* de la clase *String* o usando *StringTokenizer*, dividir la cadena en partes cada que hay una coma (,).



Pista 4: Como los identificadores no son secuenciales, es decir, no empiezan en cero y tampoco están todos los números consecutivos, una forma de manejar los vértices es usar un mapa (en Java, *HashMap* o *TreeMap*).



Ayudas para el Ejercicio 2.1



Pista 1: Usen un algoritmo para corroborar si es un grafo bipartito. Léase qué es un grafo bipartito en <http://bit.ly/2hGwAo2>



Pista 2: Si desean, pueden usar DFS o BFS para resolver este problema, pero existe otro tipo de algoritmos para resolverlo también



Pista 3: ¡Spoiler Alert! En este sitio web explican un algoritmo para verificar si un grafo es bipartito <http://bit.ly/2I0sQFZ>



Ayudas para el Ejercicio 2.2



Pista 1: Utilicen **Búsqueda en Profundidad** (*Siglas en inglés DFS*)



Pista 2: Vean Guía en **numeral 4.13** “Cómo usar Scanner o BufferedReader”



Ayudas para el Ejercicio 2.3



Pista: Usen un algoritmo para corroborar si es un grafo bipartito. Léase qué es bipartito en <http://bit.ly/2hGwAo2>



Ayudas para el Ejercicio 2.4



Pista: Algoritmos para hallar componentes fuertemente conexos. Ordenamiento topológico. DFS. Léase en <http://bit.ly/2gTeJKh>



Ayudas para el Ejercicio 3.4



Pista: Vean **Guía en numeral 4.11** “Cómo escribir la complejidad de un ejercicio en línea”



Ayudas para el Ejercicio 3.5



Errores Comunes



Ayudas para el Ejercicio 5



Pista: Para que hagan el mapa conceptual se recomiendan herramientas como las que encuentran en <https://cacoo.com/> o <https://bit.ly/2QM9D3M>



Nota 1: Si desean otra lectura, consideren la siguiente: “*John Hopcroft et al., Estructuras de Datos y Algoritmos, Capítulo 6: Grafos dirigidos. Páginas 267 – 276. 1983*” que pueden encontrarla en biblioteca



Nota 2: Estas respuestas también deben incluirlas en el informe PDF



Ayudas para el Ejercicio 6.1



Pista 1: Vean **Guía en numeral 4.21** “Ejemplo de cómo hacer actas de trabajo en equipo usando Tablero Kanban”



Ayudas para el Ejercicio 6.2



Pista 1: Vean *Guía* en numeral 4.23 “**Cómo generar el historial de cambios en el código de un repositorio que está en svn**”



Ayudas para el Ejercicio 6.3



Pista 1: Vean *Guía* en numeral 4.22 “**Cómo ver el historial de revisión de un archivo en Google Docs**”

¿Alguna inquietud?

CONTACTO

Docente Mauricio Toro Bermúdez
Teléfono: (+57) (4) 261 95 00 Ext. 9473
Correo: mtorobe@eafit.edu.co
Oficina: 19- 627

Agenden una cita dando clic en la pestaña
-Semana- de <http://bit.ly/2gzVg10>